IDerPrefs

Ian Campbell

Copyright © CopyrightÂ©1998 Ian Campbell

COLLABORATORS

	TITLE : IDerPrefs	

ACTION	*NAME*	*DATE*	*SIGNATURE*
WRITTEN BY	Ian Campbell	April 18, 2022	

REVISION HISTORY

NUMBER	DATE	DESCRIPTION	NAME

Contents

Chapter 1

IDerPrefs

1.1 IDerPrefs v2.0

```
                     IDerPrefs 2.0 A Graphical editor for "IDer" Config  ↩
                        Files
                         by Ian Campbell
                    This program is "Freeware"
```

1.2 Introduction

```
Introduction
  IDerPrefs is a graphical preferences editor for the program "IDer"
by Dave Dustin (util/wb/IDer.lha; Not in this archive).
```

Using IDer for the project icons on your system or CD-ROM has the following benefits:

- You can set the program you use for viewing any type of file once in IDerPrefs and then double clicking on the icon for a file of that type anywhere on your system will now activate that viewer.
- A bit more subtle: If you copy any project icon to type of data file it will still work.

Explanation

Programs and data have different kinds of .info files on the Amiga. Programs have "Tool" type icons, and data have "Project" type icons. The difference is, when you double-click on a "Tool" icon, the Amiga says "Oooh.. a program!" and runs it. If you double-click on a project icon, the Amiga says "Oooh.. some data, I need a program to load this into." The way it does this is that project icons have an extra bit in them which holds the name of the program to use to look at this data. Have you ever doubled-clicked on a jpeg or mod on a CD-ROM or something and your Amiga has popped up a window saying "Workbench: unable to load your tool 'modplay'" etc. What has happened is the Amiga has said "Oooh.. some data..." and found a program named to view it with in the icon and then tried to run it, but was unable to. This is because everyone has different programs installed on their system, and even if you installed all the programs you could find you still might have put them in a different directory, so it still might not be able to find them. And if you are writing a CD-ROM, how do you know what all your user's have installed? How do you make it work for everyone? This is where IDer and IDerPrefs come in.

IDerPrefs let you set for all sorts of filetypes exactly what program you want the Amiga to use when you double click on it, and exactly where to find it.

Then, when you double-click on any icon which had IDer as it's default tool, (like all the CU-Amiga CD-ROMs or any icon of your own on your system which you have changed), no more requesters. You have portable icons between all systems, and everyone has their own IDer config file on their system, telling the Amiga what viewers they personally want.

To get this to work you must first install IDer itself (util/wb/IDer.lha) then install IDerPrefs (this archive) to make editing your config file much easier.
Then you must change the default tool of all your project icons to IDer you want to be affected by IDerPrefs settings (all of them) to "IDer". This is actually easier than it sounds as there are programs for doing this on Aminet (eg. DEFT-II), so you don't have to do it manually. It is worth the effort (I think). Lastly, run IDerPrefs and configure the locations of all your programs/viewers/mod players etc.

This is not a hack at all. It is only a bit of lateral thinking by the author of IDer (not me, I wrote IDerPrefs). When IDer it the default tool and you double click the Amiga goes "Oooh, some data, I need a program to load this.. etc.)
(I promise that's the last time I'll say that :) and it goes to load the program and it's "tricked" as it is actually loading IDer and passing the name of the project to that. Then IDer uses the name to identify the file's type and then

```
load the proper tool for your system because you told it the right one with ↩
    IDerPrefs.
Of course because everyone has different settings in IDerPrefs
you can copy your icons round to your friend’s system or the reader’s on your
CD-ROM and the icons will always use their config files and their favorite ↩
    viewers.
It’s kind of like the project icons automatically change their default tool ↩
    whenever
you copy them onto someone else’s system (in-effect).  Of-course, for this to ↩
    work
both people must have IDer installed, and have made up their configs.

  The inspiration for IDerPrefs was "CUCDPrefs"  The Amiga Magazine Cover CD
Preferences Editor, which is a more specialised prefs editor for IDer.
```

1.3 Distribution

```
Distribution
IDerPrefs is Freeware.  I’m sure you’ve got better things to spend your
cash on. (Like PPC boards for instance!)
This package may be distributed via whatever means you want, provided
the archive remains intact and none of the files in it are modified.

Disclaimer
By using this product, you accept the full responsibility for any damage
or loss that might occur through its use.  I (Ian Campbell) can not be
held responsible.
```

1.4 System Requirements

```
System Requirements
- IDer 1.06 (aminet: util/wb/ider.lha)
- Kickstart 3.0 or higher
- 68020 CPU or higher
- MUI 3.8 or higher (aminet: util/libs/mui38usr.lha)
  (It may work on an earlier MUI - I don’t know, but at least 2.0 as I used
  custom classes)

If it doesn’t work on your system, please let me know.
```

1.5 Features

```
Features
- Drag & Drop a datatype icon to automatically create a new entry.
- ’Expert Mode’ allows editing of every gritty detail of config file
- ’Easy Mode’ for everyday usage.
- Partial loading of corrupt config files
- Allows a configurable "Reset to Defaults" config file
  (good for CD-ROM writers)
- Sorting of config files.
```

- MUI Interface, so customisable to your "unique" idea of style. (whatever that may be) <gulp!>
- All buttons/menus etc. work like all your other prefs editors, so you can throw this guide away now!
- Runs in Croation, Czech, Danish, English, Finnish, German, Italian, Portuguese and Serbian.
- Full English, Italian and Portuguese docs.
- Context sensitive help and bubble help.
- Now comes with a small program "AskTool". For any entry in the config file you can put this as the tool to run and then it will pop up a little window and ask you which tool to use then and there.
  Using the "Ask" checkbox in IDerPrefs will do this for you and also preserve your old tool entry in a comment, which it will restore if you click "Ask" off.

1.6 Installation

Installation

Use the installer script or just drag the IDerPrefs icon anywhere and (optionally)
set the
HELPFILE
tooltype to get online help
and the
DEFFILE
tooltype to get the Reset to Defaults option
and copy any language catalogs you want to Locale:Catalogs/<Language>.
Also, If you want to use the
Ask
option, you should also place
the program AskTool somewhere in you path. (eg. c:)

Installing IDer
If this is your first time using IDer (with or without IDerPrefs) you will have to
change the default tool of any project icons you want to be affected by your
IDer config file (whether set by IDerPrefs or by hand) to "IDer" and then place this
program somewhere in your path. (eg. copy it to C: and set the default tools to "c:IDer")
Of course this being the 1990's you wouldn't waste your time doing all your icons by
hand, and a good fast program for changing them all at once for you is DEFT-II
on Aminet. Then you should run IDerPrefs and tell it what viewers you have/want to use.

1.7 Configuration

IDerPrefs is currently only run from the Workbench and can be configured
through the following (all optional) tooltypes.

SAVENAME - This is the name/path to save the config file to

```
             when you click on 'Save'.
             Default: "envarc:IDer.prefs"

USENAME    - This is the name/path to save the config file to
             when you click on 'Use'.
             Default: "env:IDer.prefs"

DEFNAME    - This is the name/path to load the config file from
             when you chose the "Reset To Defaults" menu item.
             The default is just a config file I made in the directory,
             but this could be anything. eg. if you were using
             IDer + IDerPrefs for your data laden CD-ROM and wanted the
             user to be able to modify the viewers for their own setup,
             this would be used to point to a config file on your CD that
             you know is good, just in case they mess their one up.  Or
             a good one to use for this might be the original config file
             from the IDer archive, which I suppose you could consider
             as a default.
             Default: "progdir:def_IDer.prefs"

HELPNAME   - This is the name/path of the IDer guide file.  If
             you specify this then pressing help will load it.
             If you installed a guide file with the installer script then
             this would have been set for you.
             Default: "progdir:Docs/IDerPrefs.guide"

EXPERTMODE - =(YES|NO) Start up in the 'Expert' aka. 'The Horrible
             Truth' mode.  Lets you edit every aspect of the config file
             Default is NO. (the 'Blissfully Ignorant-I know Nozink-turn off
             the X-Files-LaLaLaLa I can't hear you' Mode)

USEIMAGEBUTTONS - =(YES|NO) If they don't suit your MUI Settings, you can
                  turn the little pictures on the Save/Use/Test/Cancel
                  buttons off with USEIMAGEBUTTONS=OFF.
                  Default: YES (use them)

CREATEICONS - =(YES|NO) If for some sick reason you like to save
             lots of little config files everywhere with 'Save As' then
             this will help you to keep track of them.
             You never get icons with the 'Save' and 'Use' buttons.
             Default: YES (to be consistent with the system editors)
```

1.8 The Easy Display

```
                The Easy Display
This is pretty easy.

The name above the listview - shows the filename of the config file
that was loaded at startup since it may have had to load the original file.
This is not the name it will be saved as when you click on save.  This is
always determined by the
              SAVENAME
               tooltype and
in it's absence it's default value (envarc:IDer.prefs).
If you want to confirm what the save name will be, pause the mouse pointer
```

over save button for a few seconds.

The Listview - Errmm.. a listview. Sorting? I have not implemented an automatic sort function here. This is because the order of the file is important, and should not just be arbitrarily rearraged (alphabetical order is also arbitrary in this context). This is because it is the order in which IDer searches for a match. You should order it yourself (using "bubble up" and "bubble down", putting your strongest and most referenced matches nearer the top. Put any weaker matches eg. Filename only ones towards the end.

The BubbleUp Arrow - swaps the currently selected entry with the one above it. You can't bubble the default (last) entry up.

The BubbleDown Arrow - swaps the currently selected entry with the one below it. You can't bubble anything down below the default (last) entry.

The Delete Button - deletes the entry currently displayed. You can delete any entry but the last one ("Default").

The Ask Button - if you select this for a file type then when you double-click on a file of that type it won't run a tool, it will pop up a window and ask you what tool to run. This is quite a good option to use permanently, say on your default config file option, that way if IDer doesn't recognise a file, it will always ask you waht to do instead of exiting silently or trying to shove it into PPMore or something. The tool editor part of the IDerPrefs display will shadow, indicating that it is no longer relevant. The tool you had there will still be saved however (IDerPrefs stores it in the IDer config file as a comment which only it recognises) so it can be restored in future if you click the "ask" option off.

Dropping a tool - on the 'EasyDisplay' bit at the bottom will set this as the action for this class of file.

Dropping a Datatype Descriptor on the listview automatically creates a new file entry, based on the file description contained in the Datatype.
See:
Technical
Save, Use, Test, Cancel - Do what they always do. In this case you can specify the save names the will be attached to these buttons.
See:
Configuration
Remember that the Save button will always save to SAVENAME and USENAME, (eg. env: as well as envarc:)

1.9 technical

Dropping a Datatype icon on the listview will parse the datatype and create a new entry for you. This is quite good as there are lots of these out there now, and you'll often get a stronger match by doing it this way

```
then by just making up your own file description (eg. #?.gif).

What the parser grabs:
  The CLASS field: is constructed by concatenating dth_Name with
  dth_GroupID eg. GIF+Picture = GIF_Picture; 8SVX + Sound = 8SVX_Sound.

  The NAME field: If the datatype descriptor contains a filename reg expn
  eg. #?.(jpg|jpeg) this will be placed in the name field.

  The OFFSET field: If the datatype descriptor contains a file Mask
  this will be stored in the offset field with "0," prepended.  Also, it will
  be stored as an ASCII mask if all of the characters in the mask are printable
  (ctype.h/isprint()) or as a hex mask otherwise.

  The FLAGS=I subfield: The value of the datatypes CaseSensitive bit
  will be used to set IDer's FLAGS=I bit. (To the opposite value, to retain
  meaning!)

  The FLAGS=T subfield: This flag is set if the datatype contains both
  a mask and a (non-#?) filetype description.  (So I assumed AND - the stricter
  the better)

  The ACTION field: (futuristic :) If the datatype contains one or more
  DTTL (Datatype Tool) chunks and one of them has a type of '3' (= editor
  tool) then this program will be straight away placed in the Action field.
  Failing that it will try to insert the type 2 (= browser) tool there.
  I do not know of any datatypes which currently use this feature, even
  the excellent akdatatypes, which would be a perfect opportunity to
  advertise SuperView I would have thought :)

  Datatypes with custom code identification hooks:
  If the datatype uses a custom comparison hook to identify it's files
  (an example of this is the MacSnd datatype) then it will be rejected
  by IDerPrefs (for obvious reasons).
```

1.10 The Expert Display

```
                The Expert Display
This is pretty easy too really.

If you've edited an IDer config file by hand before, it would be quicker
to read
              this
              .

With the expert display you get an extra bit at the bottom which lets
you edit the actual class definitions themselves.
The top half also has some extra bits too, there is an insert button now
as well as a delete button, and the Class Name text has changed to
an input string (255 characters max please).

Insert - Creates a new, blank entry for your manual editing
                  pleasure.

Stack: - The stack size for the program in Action. Default is
```

```
          4096 bytes.

Pri:    - The Amiga multitasks you know :) This is the program's
          priority, from -127 to 128. Default is 0.

No Filename Quote - Tick this if your (old) program can't parse
                    filenames in quotes, otherwise IDer will put
                    quotes around the filename to allow spaces.

Match: Cycle - IDer has the 3 ways displayed to identify a file
               (suffix, filename and character string somewhere in the
               file).  How many of these must match before it is
               satisfied enough to run the tool?

Name: - This is an AmigaDOS regular expression like the ones you
        see in the shell all the time. eg (mod.#?|#?.mod)

Suffix: - If you don't need the power of the name option, then you
          can just use this, eg. .GIF

String: - This is the most powerful option.  Did you know that most
          filetypes actually have some sort of identifier inside the
          file to say what it is.  If you know this you can enter
          it here.
          Use: <offset>, <value> for a magic number (and express it
          in hexadecimal notation eg. 16,DE..BE.. (.. is a wildcard)
          Use: <offset>, "string" if there's actually a printable
          string that's always in there that you know about.
          eg. Amigaguide documents always start with @database so you
          would put 0,@database.           (? is the wildcard here)
          "String" matches are the strongest match because it will work even
          if they name the file to something else eg. a jpeg ending in .gif
          still works!
          This field is nearly always filled in when you create your filetype
          by dragging and dropping a datatype icon.

Non Case-Sensitive - If you're using a printable string match then
                    you should really set this.
                    With this set, @Database, @dataBase, and even
                    @DaTaBaSE (eg. for some try-hard software
                    pirate's .guide file :) would count too.

Dropping a tool - on the 'Action Editor' group will make the program
                whose icon you dropped the action for this class of file.
```

1.11 Expert Display

```
I was a text file editor...
All the String Gadget's names correspond exactly to their equivalent
parameter names in the config file except:

 The "String"             contents which are put in as "OFFSET=".
 The "Non case sensitive" check activates the "I" flag.
 The "No Filename Quote"  check activates the "Q" flag.
```

The "Match:" cycle activates the "A" and "T" flags.

1.12 AskTool

What is AskTool?

AskTool is an additional and optional part of IDerPrefs introduced with v2.0. If you make AskTool the Action for a filetype when editing the text file manually, or click on "Ask" flag on when using IDerPrefs, then when you run a project of that type, no tool will be launched. Instead, a window opens up and asks you then and there which tool to use. This is very useful as the "default" option in IDerPrefs, or when you would like to try out several different viewers to see which one works best.

The AskTool Display

Here you just enter the tool you would like to use to view the file you just double clicked. If you had a tool set in IDerPrefs before you clicked on "Ask" then this will be inserted as a default. The picture button is a standard ASL file requester button. When you have the tool set up how you want it, click on "Go!".

If something goes wrong (ie. the spawned shell process returns an error state to AskTool) you will be told about it and given the chance to retry.
If all goes well then you will simply view/hear/edit your file.
If you click on the "Always use this program..." checkbox before you click "Go!" then AskTool will save your choice into the IDer.config (like clicking "Save" in IDerPrefs) so that it is always used in future,

Please note: if you ever decide to change the location where you keep your IDer.config files on your system (you can do this with the SAVENAME and USENAME tooltypes in IDerPrefs) you must run IDerPrefs after this, and click save once (no need to change anything). This will inform AskTool of the change you have made. (If you look at the config file in a text editor you will understand why .)
This little bit of work when you change your config file location (if ever) allows
AskTool to be permanently a purely self contained program with no need for an icon/tooltypes or any environment variables.

1.13 Contacting the Author

Contact Info

EMail:
icampbell@xtra.co.nz

About me

I am a 23 year old male living in Wellington on the North Island of New Zealand, where the air is clean and fresh and there are lots of snowy mountains and stuff.

I work as an Analyst/Programmer for a legal publisher, involving work with SGML and document processing languages such as Perl & Omnimark under a Windows '95 environment. I genuinely enjoy programming of all types and it was a great pleasure for me to write IDerPrefs which is why it is free. And, for those who are wondering, IDerPrefs is not a Perl script, it's an Object Oriented C++ program :)

```
    I still think the Amiga is a great home computer, and has a lot of
  potential (which it looks like Gateway 2000 have noticed)!
    I have been a dedicated Amigan since I sold my C64 in 1989
  and brought an A500.  My current system is an A1200 Blizzard ’030/50,
  16+2 RAM, 6xCDROM, C=1942 Monitor and two 2.5" drives crammed in there -
  a 500Mb and a 60Mb. (Perhaps "the" average Amiga system these days or is an
  ’040 pretty much standard now?)

  Enjoy the program fellow Amigans!
```

1.14 Thanks to y’all

```
 Thanks to the authors of the following:
 - "IDer 1.06" by Dave Dustin (of course :)  (aminet: util/wb/IDer.lha)
   (Sorry, I couldn’t contact you about "IDerPrefs", your email address must have ↩
      changed -
    - this is why I left IDer out of the archive.)
 - "SAS/C 6.58" from the SAS Institute and Steve Krueger.
 - "MUI 3.8" by Stefan Stuntz (very clever). (aminet: util/libs/mui38demo.lha)
 - "Localizer 1.37" by Massamilliano Origgi which made localising a breeze.
    (aminet: dev/misc/Localizer1.37)
 - "MakeDT 1.3" by Michal Letowski, whose AREXX source code allowed me to
   write a parser for the those little datatypes files in
   devs:datatypes. (aminet: dev/misc/MakeDT-1.3.lha)
 - The maintainers of Aminet
 - All those involved with ATO - The Amiga Translators Organisation who translated
   the program strings.
 - Davide Romanini and Francesco Celli from ATO for not only translating the  ↩
    strings, but
   also the docs into Italian (brilliant!).
 - Joel and Rúben Alvim for doing full Portuguese Docs (Great!).
```

1.15 Program History

```
Version 1.0  (04.02.1998)
- First Release

Version 1.1  (18.02.1998)
- Name changed to "ViewerPrefs"
- German, Italian and Serbian language catalogs added. (ATO)
- Italian docs added. (ATO)
- Installer script changed to handle languages.
- BUGFIX: If the icon was renamed or deleted the program would crash.
  Now it asks wb for the name.
- BUGFIX: Config file entries with blank action fields would be
  overwritten by the following entry by the loader.
- BUGFIX: Display would screw up when changing from expert back to
  easy.
- NEW: ’Test’ button added.
- NEW: Datatype entry creator much improved and bugs fixed.
- NEW: Now compiled for 68020 for more speed. (Data was longword aligned in 1.0  ↩
   anyway)
```

```
- NEW: Archive now contains large pre-done 53-entry data file.
- CHANGE: ORIGNAME tooltype changed to the more logical DEFNAME.
- CHANGE: English guide file has new intro/history and names for IDerPrefs.
- CHANGE: Changed "IDerPrefs" strings in Italian guide to "ViewerPrefs" for the  ↩
   new version.
- CHANGE: New MagicWB & NewIcons icons.

Version 1.2  (02.03.1998)
- NEW: Slider replaced with an MUI Listview (Idea:Gero Busse)
- NEW: You can now sort the list
- Added Croation, Danish and Portuguese languages. (ATO)
- Added Portuguese docs. (ATO)
- Updated installer script to handle to new languages and guide.
- Updated English guide to include description of bubble buttons and history.
- Sorted the large demo config with the new sort feature.

Version 1.3  (07.03.1998)
- BUGFIX: Class Name wasn't updated when dropping a datatype icon on the listview.
- BUGFIX: Action Editor heading string was mixed up in English Locale.
- Small Graphical improvements.
- New Help bubbles (English only)
- Added MPEG Audio, MPEG Video and WAVE Sound to default config.
- IDerPrefs now defaults to expert mode.

Version 1.3a (15.03.1998)
- Added Czech translation.

Version 2.0  (09.05.1998)
- Name changed back to IDerPrefs. Ahem... ;)
- New 'Ask' option for the filetype action.  If you activate this for a filetype,  ↩
   then
  double click the icon for one of these files, a little window will pop up asking ↩
      you
  what program you would like to view it with.
- Added a workaround to re-activate multilingual Save/Use/Cancel buttons lost in  ↩
   1.1.
- Added Czech catalog to installer script.  Forgot in 1.3a, sorry.
- Added full support for spaces in class name descriptions, eliminating the need  ↩
   for
  all those underscore characters. Class name descriptions, suffix and name  ↩
     entries
  now implicity have (are saved with) quotes around them (this is OK with IDer),
  so spaces are allowed in them. The only place you should explicity use quotes  ↩
     now
  (and the only place you're allowed to - thanks to MUI) is in the "String" field  ↩
     if you
  want to do a text (as opposed to data) match.
- Recompiled with SAS/C 6.59. Program size reduced by 46,000 bytes
  (and is therefore presumably faster)
- Easy Mode is the default again.
- Added the AskTool program to the archive and installer script.
- Removed the picture of the wizard.  It was cute, but redundant.  Just drop a  ↩
   datatype
  descriptor on the listview to add a new entry to it (more intuitive).
- There is now an IMAGEBUTTONS tooltype for activating or deactivating the image  ↩
   buttons
  introduced in 1.3
```

```
- BUGFIX: Removed an enforcer hit on insertion (gotta watch what those MUI notifications
  are doing!) which has been there since at least 1.3 and cleaned up insertion in general.
- Now uses MUI's shadowing feature to show when buttons are disabled, instead of them
  just doing nothing.  Eg. deleting/moving the default entry from the end of the file,
  and others.
- Default Config file expanded and more descriptive names
  (select "Reset to Defaults" to see the new default file)
- Reworked config file parser.

Version 2.0a (10.05.1998)
- BUGFIX: AskTool tripped up on filenames with spaces.  Fixed.
```